

In this lab, we will learn to use:

1. Data encapsulation
2. Inheritance
3. Polymorphism

The surface-area-to-volume ratio is an important concept in science and engineering. It is used to explain the relation between structure and function in processes occurring through the surface and the volume. One such example is diffusion and heat transfer by conduction.

In this lab, we will compare the surface-area-to-volume ratio of several shapes. The code for this is already written in the ClassTesterProgram.py, which you can download from D2L. However, none of the classes have been written. You need to implement the following class diagram.

Class diagram... (See below)

Requirements & Hints:

- Use Inheritance and Polymorphism to ensure subclasses have the needed methods.
You could likely complete this lab without Inheritance or Polymorphism... Don't do that!
- The Square class only needs a constructor.
- In the circle class, the two helper functions *calcAreaOfCircle* and *calcPerimeterOfCircle*. The *getArea* and *getPerimeter* methods should use these two helper functions. Additionally, these helpers should only be available to the circle class and any subclasses... Think Public, Protected and Private. In Python, the syntax for protected is *_methodName* and the syntax for private is *__methodName*
- If you don't know a formula, Google it.
- Please make comments in your code where you are using data encapsulation, inheritance, and Polymorphism

Once you're done, upload your .py file with your class definitions to D2L.





